October 2003

Visual Servoing of a UGV from a UAV using Differential Flatness

Rahul Rao
University of Pennsylvania

R. Vijay Kumar
University of Pennsylvania, kumar@grasp.upenn.edu

Camillo J. Taylor
University of Pennsylvania, cjtaylor@cis.upenn.edu

Recommended Citation

Rao, Rahul; Kumar, R. Vijay; and Taylor, Camillo J., "Visual Servoing of a UGV from a UAV using Differential Flatness" (2003). *Departmental Papers (MEAM)*. 20.
http://repository.upenn.edu/meam_papers/20

Copyright 2003 IEEE. Reprinted from *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Sytems (IROS 2003)*, Volume 1, pages 743-748.
Publisher URL: http://ieeexplore.ieee.org/xpl/tocresult.jsp?isNumber=27983&page=8

Visual Servoing of a UGV from a UAV using Differential Flatness

Abstract
In this paper the problem of controlling the motion of a nonholonomic vehicle along a desired trajectory using observations from an overhead camera is considered. The control problem is formulated in the image plane. We show that the system in the image plane is differentially flat and use this property to generate effective control strategies using only visual feedback. Simulation results illustrate the methodology and show robustness to errors in the camera calibration parameters.

Proceedings of the 2003 IEEE/RSJ
Intl. Conference on Intelligent Robots and Systems
Las Vegas, Nevada · October 2003

Visual Servoing of a UGV from a UAV using Differential Flatness

Rahul Rao* Vijay Kumar* Camillo Taylor†
General Robotics Automation, Sensing and Perception (GRASP) Laboratory
* Department of Mechanical Engineering and Applied Mechanics
† Department of Computer and Information Science
University of Pennsylvania, 3401 Walnut Street, Philadelphia, PA 19104
E-mail: {rahulrao, kumar, cjtaylor}@grasp.cis.upenn.edu

***Abstract*— In this paper the problem of controlling the motion of a nonholonomic vehicle along a desired trajectory using observations from an overhead camera is considered. The control problem is formulated in the image plane. We show that the system in the image plane is differentially flat and use this property to generate effective control strategies using only visual feedback. Simulation results illustrate the methodology and show robustness to errors in the camera calibration parameters.**

I. Introduction

Visual servoing, i.e. the use of feedback from a camera or a network of cameras, has been used increasingly in the last few years for the development of control algorithms such as motion control for mobile robots [7], robot manipulation tasks [11] and remote surveillance tasks [13]. Visual servoing is particularly relevant for the control of *Eye-in-hand* systems in manipulation tasks. The work in this area can be classified into one of two approaches — *position based* and *image based* control systems [1], [6]. In a *position based* (or *3D visual servoing*) control system features are extracted from the image and used in conjunction with a geometric model of the target and the known camera model to estimate the target's pose with respect to the camera. Control inputs are computed based on the errors in the estimated pose space thus making camera calibration necessary for reliable control. In the *image based* (or *2D visual servoing*) approach, controls are computed directly on the basis of image features, thus reducing errors due to sensor modeling and camera calibration. However the design of the controller in this case is harder since the plant is highly nonlinear [6]. Taking advantage of *2D* and *3D* visual servoing techniques is the approach called *2-1/2 D* visual servoing. This approach is claimed to be more robust with respect to calibration errors, though it is also more sensitive to image noise if image features are used to compute control inputs [8].

In contrast to the papers mentioned above which mostly deal with kinematic models, visual feedback can also be combined with systems with second order dynamics. Ma, Košeçká and Sastry [7] used visual servoing techniques to control the motion of a car based on information obtained from a camera mounted on the vehicle. Zhang and Ostrowski's [13] used visual servoing techniques for the control of an unmanned blimp. Finally, Cowan and Koditschek [3] used navigation functions on the image plane to design image-based servo algorithms to guide a planar convex rigid body to a static goal for all initial conditions within the camera's workspace.

This paper is motivated by applications where one may want to control a UGV or a fleet of UGVs from overhead cameras, possibly mounted on UAVs. This is in contrast to the work discussed above in which the camera is physically attached to the robot. Specifically, we address the visual servoing of car like robots from an overhead camera that may be mounted on a UAV. We exploit the well-known property of *differential flatness* for nonholonomic car-like robots [10]. Roughly speaking, any system is differentially flat if one can find a set of outputs (which is equal in number to the set of inputs) such that all the states of the system and all the inputs for the system can be expressed in terms of these outputs and their higher derivatives. In other words, all the states and inputs of the system can be expressed in terms of the flat outputs without integration. In our system, the inputs are the usual angular and linear velocities for the robot but the feedback is provided via the positions of a reference point on the car seen in the image of an overhead camera. We show that the system is differentially flat with the flat outputs being the image coordinates of the reference point. We show that the visual-servoing problem can be easily solved using this framework and provide simulation results to illustrate our methodology.

We briefly describe the visual servoing problem and the paradigm of differential flatness in Section II. We establish in Section III that the system dynamics, when transferred to the image plane, possesses the property of differential flatness. Thus, a controller can be designed in the flat space using methods from linear systems theory and then mapped back into the real world. These ideas are demonstrated in Section IV. Some results from simulation both for systems with and without parametric uncertainties are presented in Section V. Finally, Section VI discusses the advantages and limitations of our approach and our

0-7803-7860-1/03/$17.00 © 2003 IEEE

future work in this area.

II. Background

This paper addresses a strategy for image- based control of the position and orientation of a moving vehicle (or vehicles) based on images obtained from remotely located vehicles. Figure 1 shows an example of such a situation.

In Figure 1, the camera **C** is attached to an overhead blimp such that it is able to see two objects on the ground, namely the robot and its target. The robot kinematics are subject to non- holonomic constraints. It has no on-board cameras that enable it to *see* the world around it. In this framework of motion planning and control, a controller is designed for the robot in the image plane and these controls are then transferred into the world frame in order to enable the robot to follow the desired trajectory. In Section III we show that the visual servoing problem can be formulated as a differentially flat system.

The basic idea of a differentially flat system was introduced by Fliess et. al [5]. A differentially flat system is suited for trajectory generation tasks [12] including real time feasible trajectory generation in the presence of inequality constraints [4]. Mathematically, a nonlinear system

$$\dot{x} = f(x,u) \quad (1)$$

$$y = h(x) \quad (2)$$

is differentially flat if we can find flat outputs

$$z_f = z_f(x, u, \dot{u}, \ddot{u}, \cdots, u^{(p)}) \quad (3)$$

such that

$$x = x(z_f, \dot{z}_f, \ddot{z}_f, \cdots, z_f^{(p)}) \quad (4)$$

$$u = u(z_f, \dot{z}_f, \ddot{z}_f, \cdots, z_f^{(p)}) \quad (5)$$

z_f is called the flat output which might or might not be the same as y, the tracking output. As represented above in Eqs. (4) and (5), the states and the input variables of the system are expressed in terms of the flat outputs and their higher derivatives. Murray et. al [10] provide many examples of mechanical systems that are differentially flat. As shown by Nieuwstadt and Murray [12], it is relatively simple to generate trajectories for differentially flat systems, even when there are inequality constraints on the state [4].

III. Visual Servoing and Differential Flatness

A. Imaging Model

Let $\mathbf{w} \equiv (x, y, 1)^T$ denote the homogeneous coordinates of a point on the ground plane and $\mathbf{c} = (u, v, 1)^T$ denotes the coordinates of the projection of $\mathbf{w}$ in the image. It is easy to show that $\mathbf{w}$ and $\mathbf{c}$ are related by a projective transformation $\mathbf{G}$. This can be expressed as

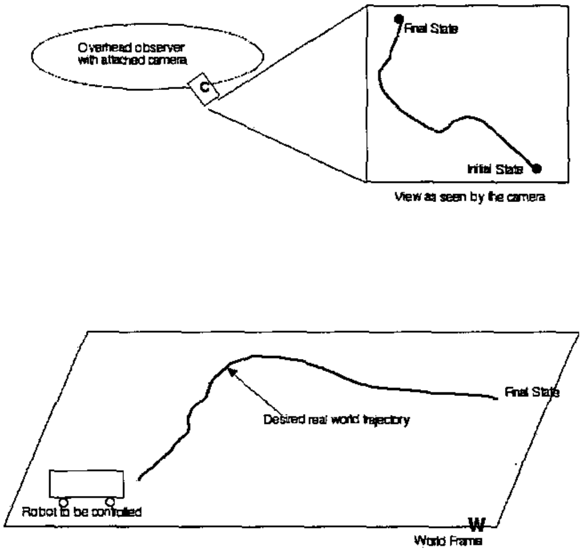
Overhead observer with attached camera
C
Final State
Initial State
View as seen by the camera
Desired real world trajectory
Final State
Robot to be controlled
W
World Frame

Fig. 1. A representation of a vehicle being guided along a desired path by a remote observer

$$\mathbf{c} \propto \mathbf{G}\mathbf{w},\ \mathbf{G} \in GL(3) \quad (6)$$

$$\Rightarrow \mathbf{w} \propto \mathbf{H}\mathbf{c} \quad (7)$$

where $\mathbf{H} = \mathbf{G}^{-1}$.

Let the matrices $\mathbf{G}$ and $\mathbf{H}$ be represented in terms of their columns as $\mathbf{G} = \begin{pmatrix} G^1 & G^2 & G^3 \end{pmatrix}$ and $\mathbf{H} = \begin{pmatrix} H^1 & H^2 & H^3 \end{pmatrix}$ respectively. Similarly, let the rows of $\mathbf{G}$ and $\mathbf{H}$ be represented as $\begin{pmatrix} G_1 & G_2 & G_3 \end{pmatrix}^T$ and $\begin{pmatrix} H_1 & H_2 & H_3 \end{pmatrix}^T$ respectively. Replacing the proportionality signs in (7) by equalities we get

$$\mathbf{w} = \lambda \mathbf{H}\mathbf{c} \quad (8)$$

where $\lambda = \frac{1}{H_3.\mathbf{c}}$. Similarly, from (6) we obtain

$$u = \frac{G_1.\mathbf{w}}{G_3.\mathbf{w}} \quad (9)$$

$$v = \frac{G_2.\mathbf{w}}{G_3.\mathbf{w}} \quad (10)$$

Differentiating (9) with respect to time yields

$$\dot{u} = \frac{(G_3.\mathbf{w})(G_1.\dot{\mathbf{w}}) - (G_1.\mathbf{w})(G_3.\dot{\mathbf{w}})}{(G_3.\mathbf{w})^2}. \quad (11)$$

Using the expression for $\mathbf{w}$ from (8) we get

$$\dot{u} = \frac{(G_3.(\lambda\mathbf{H}\mathbf{c}))(G_1.\dot{\mathbf{w}}) - (G_1.(\lambda\mathbf{H}\mathbf{c}))(G_3.\dot{\mathbf{w}})}{(G_3.(\lambda\mathbf{H}\mathbf{c}))^2}. \quad (12)$$

From (6) and (7) it can be shown that $G_3.(\mathbf{H}\mathbf{c}) = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \cdot \mathbf{c} = 1$. Using this fact, the expression for λ, and simplifying (12) we get

$$\dot{u} = (H_3 \cdot \mathbf{c})[G_1.\dot{\mathbf{w}} - u(G_3.\dot{\mathbf{w}})] \tag{13}$$

Similarly

$$\dot{v} = (H_3 \cdot \mathbf{c})[G_2.\dot{\mathbf{w}} - v(G_3.\dot{\mathbf{w}})] \tag{14}$$

These expressions can be written compactly as

$$\begin{pmatrix}\dot{u}\\ \dot{v}\end{pmatrix} = (H_3 \cdot \mathbf{c})\begin{pmatrix}G_1.\dot{\mathbf{w}} - u(G_3.\dot{\mathbf{w}})\\ G_2.\dot{\mathbf{w}} - v(G_3.\dot{\mathbf{w}})\end{pmatrix} \tag{15}$$

$$= (H_3 \cdot \mathbf{c})\begin{pmatrix}1 & 0 & -u\\ 0 & 1 & -v\end{pmatrix}\mathbf{G}\dot{\mathbf{w}} \tag{16}$$

where $\dot{\mathbf{w}} = (\dot{x}, \dot{y}, 0)^T$. This can be rewritten as

$$\begin{pmatrix}\dot{u}\\ \dot{v}\end{pmatrix} = (H_3 \cdot \mathbf{c})\begin{pmatrix}1 & 0 & -u\\ 0 & 1 & -v\end{pmatrix}\begin{pmatrix}G^1 & G^2\end{pmatrix}\begin{pmatrix}\dot{x}\\ \dot{y}\end{pmatrix} \tag{17}$$

Note that H_3, the third row of $\mathbf{H}$, can be expressed in terms of the columns of $\mathbf{G}$ as

$$H_3 = \frac{G^1 \times G^2}{(G^3) \cdot (G^1 \times G^2)} \tag{18}$$

Also, it must be noted that $\frac{1}{\lambda} = (H_3 \cdot \mathbf{c})$ is a scalar that varies with u and v. The expression for image frame velocities, $\dot{u}$ and $\dot{v}$, given in Equation 17 is bilinear in u and v and linear in $\dot{x}$ and $\dot{y}$.

B. *The Vehicle Model*

The kinematic car or the unicycle is represented by

$$\begin{aligned}\dot{x} &= u_1 \cos\phi\\ \dot{y} &= u_1 \sin\phi\\ \dot{\phi} &= u_2\end{aligned} \tag{19}$$

where u_1 and u_2 are the forward and angular velocities of the robot respectively. It can be shown to be a differentially flat system with the coordinates of the center of the rear axle, $(x, y) \equiv (y_1, y_2)$, being the flat outputs. We next show that if the image coordinates of the center of the rear axle, (u, v), are chosen as the flat outputs, we can express the states of the system, (x, y, ϕ) and the system inputs, (u_1, u_2) in terms of (u, v), $(\dot{u}, \dot{v})$ and $(\ddot{u}, \ddot{v})$. This enables us to transfer the original control problem into the image plane, obtain a flat space representation for the control problem in the image plane, solve the trajectory tracking control problem in the flat space using linear control techniques and then use the inverse mapping to obtain real world control inputs to enable the robot to track the desired trajectory.

To establish the property of differential flatness, first notice that the states (x, y) can be expressed in terms of (u, v) using (8). Next, note that (17) can be written in a drift-free form in the image plane as

$$\begin{pmatrix}\dot{u}\\ \dot{v}\end{pmatrix} = \mathbf{F}\begin{pmatrix}\dot{x}\\ \dot{y}\end{pmatrix} \tag{20}$$

where

$$\mathbf{F} \equiv \frac{1}{\lambda}\mathbf{M}\begin{pmatrix}G^1 & G^2\end{pmatrix} \tag{21}$$

and

$$\mathbf{M} = \begin{pmatrix}M_1\\ M_2\end{pmatrix} = \begin{pmatrix}1 & 0 & -u\\ 0 & 1 & -v\end{pmatrix} \tag{22}$$

Thus,

$$\begin{pmatrix}\dot{x}\\ \dot{y}\end{pmatrix} = \mathbf{F}^{-1}\begin{pmatrix}\dot{u}\\ \dot{v}\end{pmatrix} \tag{23}$$

This enables us to write the third state ϕ as the ratio of the expressions for $\dot{x}$ and $\dot{y}$. Specifically,

$$\tan\phi = \frac{\dot{u}(-M_2G^1) + \dot{v}(M_1G^1)}{\dot{u}(M_2G^2) - \dot{v}(M_1G^2)} \tag{24}$$

The inputs u_1 and u_2 can also be expressed in terms of u, v and their higher derivatives (upto the second order). Squaring and adding the expressions for $\dot{x}$ and $\dot{y}$, we get

$$u_1 = \frac{\delta(u, v, \dot{u}, \dot{v})}{\sigma(u, v)} \tag{25}$$

where

$$\delta^2 = (\dot{u}(-M_2G^1) + \dot{v}(M_1G^1))^2 + (\dot{u}(M_2G^2) - \dot{v}(M_1G^2))^2 \tag{26}$$

and $\sigma(u, v)$ is a nonzero polynomial term. Similarly, differentiating (24) and simplifying the resulting expression gives us an expression for the other input $u_2 = \frac{\chi(u,v,\dot{u},\dot{v},\ddot{u},\ddot{v})}{u_1^2}$. Thus, the variables (x, y, ϕ, u_1, u_2) are expressed in terms of the flat outputs (u, v) and their higher derivatives, except at $u_1 = 0$. Thus the system given by (9, 10, 19) is differentially flat, except for the singularity at $u_1 = 0$.

IV. The Control Scheme and Simulations

As a consequence of the previous flatness result, the system is fully linearizable by dynamic feedback [9] in the image plane. Differentiating (17) with respect to time once again so that the second input u_2 also appears in the dynamics of the system, we obtain

$$\begin{pmatrix}\ddot{u}\\ \ddot{v}\end{pmatrix} = \mathbf{T}\begin{pmatrix}\dot{u_1}\\ u_2\end{pmatrix} + \begin{pmatrix}\alpha_1\\ \alpha_2\end{pmatrix} \equiv \begin{pmatrix}v_1\\ v_2\end{pmatrix} \tag{27}$$

or

$$\begin{pmatrix}\ddot{u}\\ \ddot{v}\end{pmatrix} = \begin{pmatrix}v_1\\ v_2\end{pmatrix} \tag{28}$$

where $\mathbf{v} = (v_1, v_2)^T$ are the inputs to the system in the flat space. $\mathbf{T}$ is a (2×2) matrix whose elements, like α_1 and α_2, are scalars dependent on the position, (u, v), of the robot in the image, the calibration matrix $\mathbf{G}$, the input u_1 and the orientation ϕ of the robot in the real world, as obtained from (24). The system (28) can further be written in a compact form as

$$\dot{\mathbf{z}} = \mathbf{A}\,\mathbf{z} + \mathbf{B}\,\mathbf{v} \tag{29}$$

where $\mathbf{z} \equiv (z_1, z_2, z_3, z_4)^T = (u, \dot{u}, v, \dot{v})^T$. Further, the structure of the system (29) looks like

$$\begin{aligned} \dot{z}_1 &= z_2 \\ \dot{z}_2 &= v_1 \qquad (30) \\ \dot{z}_3 &= z_4 \\ \dot{z}_4 &= v_2 \qquad (31) \end{aligned}$$

Thus, given a desired trajectory of the form $(z_{1d}(t), z_{3d}(t))$ in the flat space, one can choose the controller in the flat space to be of the form

$$\begin{aligned} v_1(t) &= \ddot{z}_{1d}(t) - k_1(\dot{z}_{1d}(t) - \dot{z}_1(t)) - k_0(z_{1d}(t) - z_1(t)) \\ v_2(t) &= \ddot{z}_{3d}(t) - k_3(\dot{z}_{3d}(t) - \dot{z}_3(t)) - k_2(z_{3d}(t) - z_3(t)) \quad (33) \end{aligned}$$

From (17) it can be observed that $(\dot{z}_{1d}(t), \dot{z}_{3d}(t)) \equiv (\dot{u}_d(t), \dot{v}_d(t))$ can be expressed as

$$\begin{pmatrix} \dot{u}_d \\ \dot{v}_d \end{pmatrix} = (H_3 \cdot \mathbf{c_d}) \begin{pmatrix} 1 & 0 & -u_d \\ 0 & 1 & -v_d \end{pmatrix} \begin{pmatrix} G^1 & G^2 \end{pmatrix} \begin{pmatrix} \dot{x}_d \\ \dot{y}_d \end{pmatrix} \tag{34}$$

where $(x_d(t), y_d(t))$ is the *desired real world* trajectory, and $\mathbf{c}_d = (u_d(t), v_d(t), 1)$ is the homogeneous *desired image* coordinates. Once these controls are chosen, (27) can be used to compute the controls in the real world, i.e. the controls, in terms of forward and angular velocities that will be required to drive the robot along a desired trajectory. Specifically, it leads to

$$\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \mathbf{T}^{-1} \begin{pmatrix} v_1 - \alpha_1 \\ v_2 - \alpha_2 \end{pmatrix} \tag{35}$$

It must be noted, however, that while the control laws (32) involve the states (z_1, z_2, z_3, z_4), only the states (z_1, z_3) are directly measured by the camera. Thus, in order to use these controllers to control the robot's motion a partial state observer or a full state observer will also have to be designed in order to estimate the states z_2 $(= \dot{z}_1)$ and z_4 $(= \dot{z}_3)$. Having transformed the problem from its original nonlinear structure to a linear framework, the task of designing a linear state observer in the flat space can be accomplished using linear system theory. The rate of convergence of the estimator can be controlled so that the estimated state approaches the actual state at a desired rate by a proper choice of the observer gain matrix [2].

V. Simulation Results

In this section we present some results obtained by simulating the controller described earlier controlling the motion of a robot along desired trajectories. Though the desired trajectory is specified (such as a circle or an ellipse or a path generated by a motion planner), the starting position and orientation of the robot in the real world is chosen arbitrarily, the only constraint being that it should lie within the camera's field of view. The full state observer described briefly above is used to establish the state vector.

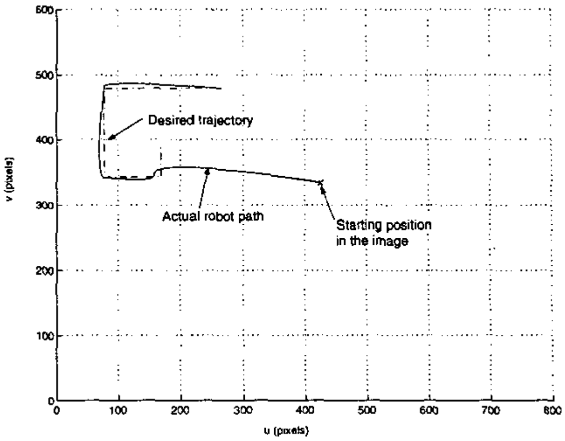

Fig. 2. The robot path as seen in the image (solid) and the desired image plane trajectory

Figure 2 shows the path traced by the robot, as seen in the image, under the influence of the controller designed using the procedure described earlier. The starting position of the robot is indicated by a cross-hair and it can be seen that the robot does not start on the desired trajectory but converges to the desired trajectory.

Another example is shown in Figure 3. In this case too, the robot's initial position and orientation in the real world are indicated by the line and the cross- hairs. The forward and angular velocities of the robot as computed by the estimator are shown in Figure 4 for the time that the robot is in motion.

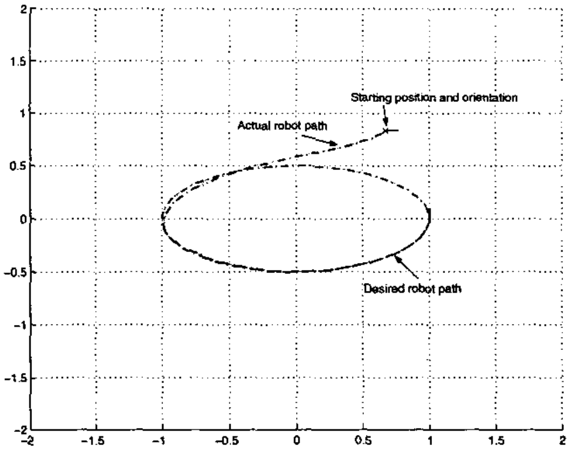

Fig. 3. The robot path (dot-dash) and the desired elliptical trajectory in the real world

By introducing uncertainties in the calibration matrix we can analyze the robustness of the visual feedback con-

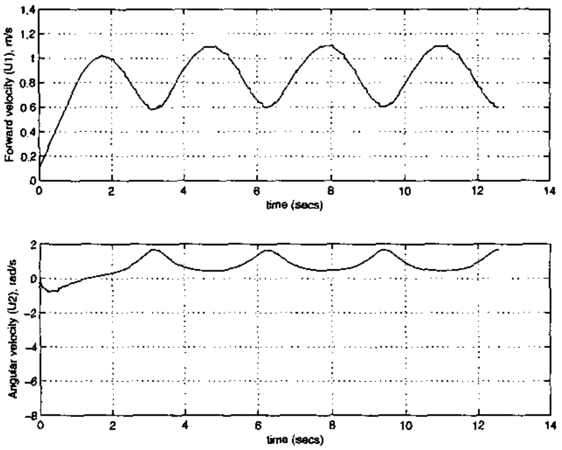

Fig. 4. The forward and angular velocities of the robot while tracking the desired trajectory in Fig. 3

troller. We simulate uncertainties by errors in the intrinsic parameters such as the camera focal length and the offsets in the retinal coordinate system. Figure 5 shows the path traced by the robot as seen in the image along with the desired path. The figure shows that the controller tries to track a desired trajectory inspite of a 10% uncertainty in these intrinsic parameters.

Simulations were also carried for robot control in the presence of motion of the camera with respect to the ground plane. The camera motion was characterized by rotation of 10° in camera orientation (relative to the camera's *original* frame with respect to the world) caused by rotations about random unit vectors $\overline{\eta}$ generated continuously during simulation while the robot was moving. Figure 6 shows the results of a sample experiment which illustrates controller convergence to the desired path despite camera motion.

VI. Discussion and Future Work

We have formulated the problem of controlling a mobile robot with feedback from overhead imagery that can be obtained from a UAV. The resulting system is shown to be differentially flat, with the flat outputs being the coordinates of a reference point on the image plane. This allows us to formulate the trajectory generation and control problem in the image space, and leverages the results of ([4], [5], [12]) on differentially flat systems. Further simulation results show that the system performs reasonably well in the presence of uncertainties. For example, Figure 5 shows reasonably good performance even when the calibration matrix is inaccurate, while Figure 6 shows that the path traced by the robot is fairly good inspite of errors in the estimate for the overhead camera's orientation, as described in Section V.

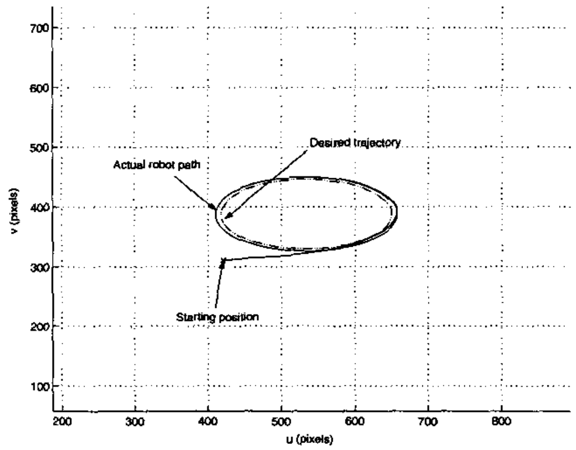

Fig. 5. The robot path *solid* as seen in the image and the desired image plane trajectory in the presence of calibration matrix uncertainties

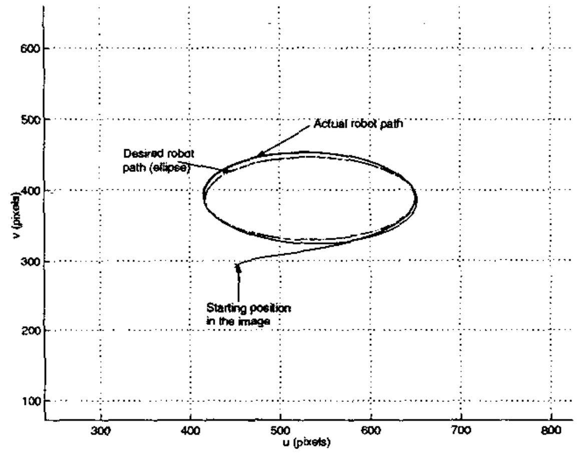

Fig. 6. The robot path (solid) as seen in the image and the desired image plane trajectory in the presence of uncertainties in the extrinsic parameter matrix

While simulation results do appear to be encouraging, it is important to note that we have not explicitly accounted for camera motion. A camera on the UAV will be subject to continuous rotations and translations. This will not only result in unknown calibration parameters, but in addition, the calibration parameters will change with camera motion. Even if we assume that the dynamics of the camera motion is slow compared to the dynamics of the UGV, it is necessary to develop an online scheme that will enable the estimation of the homography, **G**. Our future work involves the use of this paradigm in a multi-robot setting, where estimates of relative positions of UGVs can be shared with the UAV to provide the required information for estimating **G**.

Acknowledgement

This work was in part supported by NSF grant IIS00-83240 and ARO MURI grant DAAD19-02-01-0383.

VII. REFERENCES

[1] In K. Hashimoto, editor, *Visual Servoing: Real Time Control of Robot Manipulators Based on Visual Sensory feedback.* World Scientific Series in Robotics and Automated Systems, 1993.

[2] C.-T. Chen. *Linear System Theory and Design.* Oxford University Press, 1999.

[3] N. J. Cowan and D. E. Koditschek. Planar image based visual servoing as a navigation problem. In Proc. IEEE Int. Conf. Robotics and Automation, pages 611–617, May 1999.

[4] N. Faiz, S. K. Agrawal, and R. M. Murray. Trajectory planning of differentially flat systems with dynamics and inequalities. *AIAA Journal of Guidance, Navigation and Control*, 24(2):219–227, 2001.

[5] M. Fliess, J. Levine, and P. Rouchon. Flatness and defect of nonlinear systems: Introductory theory and examples. *International Journal of Control*, 61(6):1327–61, 1995.

[6] S. Hutchinson, G. Hager, and P. Corke. A tutorial on visual servo control. *IEEE Trans. on Robotics and Automation*, May 1996.

[7] Y. Ma, J. Košecká, and S. Sastry. Vision guided navigation for a nonholonomic mobile robot. *Trans. on Robotics and Automation*, 15(3):521–536, June 1999.

[8] E. Malis, F. Chaumette, and S. Boudet. 2 1/2 d visual servoing. *Trans. on Robotics and Automation*, 15(2):238–250, April 1999.

[9] P. Martin, R. M. Murray, and P. Rouchon. Flat systems. in Plenary lectures and mini courses, European Control Conference, July 1997.

[10] R. M. Murray, M. Rathinam, and W. Sluis. Differential flatness of mechanical control systems: A catalog of prototype systems. In *Proc. ASME Int. Mech. Eng. Congress and Expo*, San Francisco, November 1995.

[11] C. J. Taylor, J. P. Ostrowski, and S. H. Jung. Robust visual servoing based on relative orientation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 574–580, Fort Collins, CO, June 1999.

[12] M. J. van Nieuwstadt and R. M. Murray. Real time trajectory generation for differentially flat systems. *International Journal of Robust and Nonlinear Control*, 8, No. 11:995–1020, 1998.

[13] H. Zhang and J. P. Ostrowski. Visual servoing with dynamics: Control of an unmanned blimp. In *Proc. Int. Conf. on Robotics and Automation*, pages 618–623, Detroit, May 1999.